Indian Institute of Technology Patna
CS1101- Foundations of Programming

Lab-1 : Introduction to Linux

Date: 4-8-2025

1.1 What is Linux?

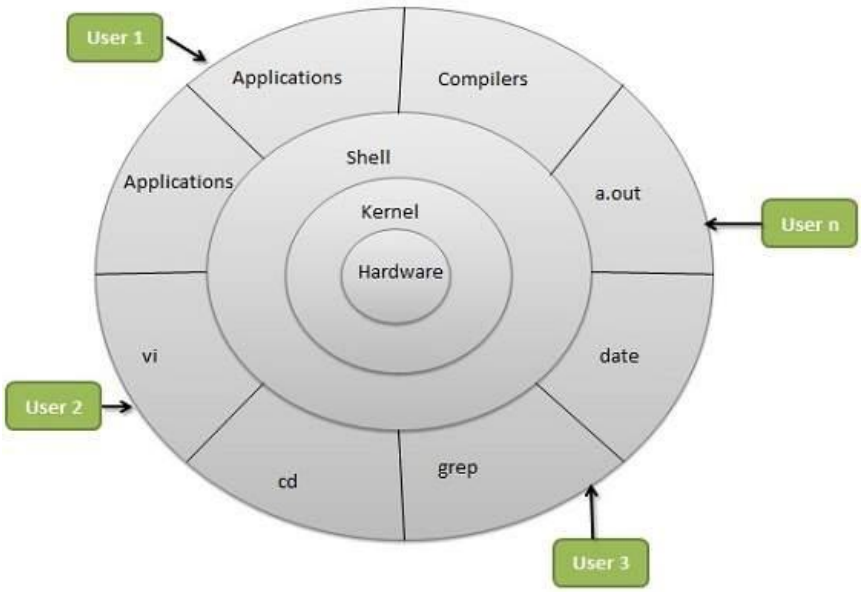
Linux is a free, open-source, Unix-like operating system kernel originally developed by Linus Torvalds in 1991. Today, the term "Linux" often refers to Linux-based operating systems (called distributions) that include the Linux kernel along with system utilities, libraries, and applications.

1.2 Why Use Linux?

- Free and open-source.
- Stability and security.
- Preferred in servers, development, embedded systems.

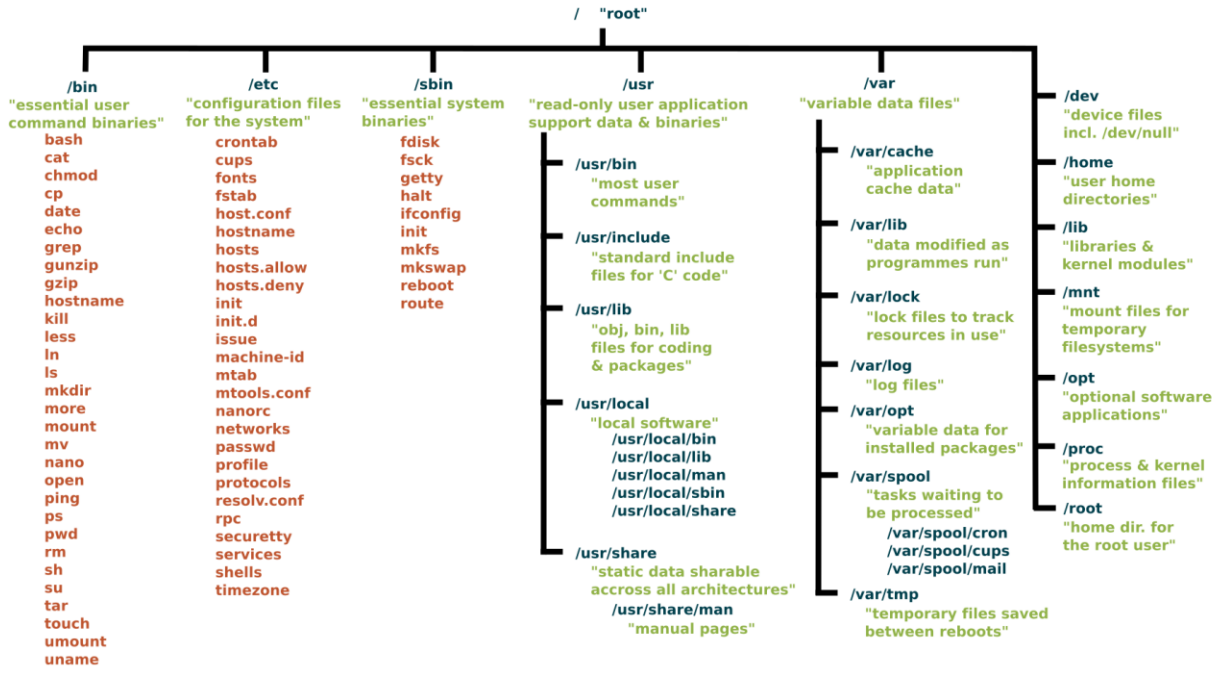
1.3 Linux Architecture

- Kernel: Core OS functionality
- Shell: Interface between user and OS
- File System: Hierarchical structure
- Userspace: Applications, system tools



Linux Filesystem Structure

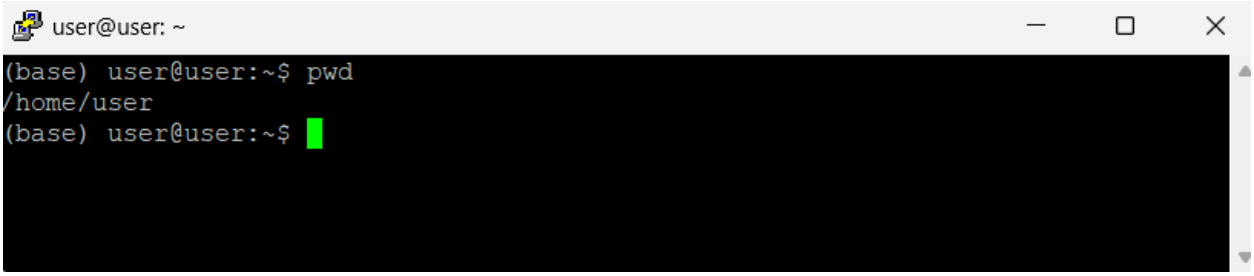
Directory	Description
/	Root directory
/home	User directories
/etc	Configuration files
/bin	Essential binaries
/usr	User-installed software
/var	Logs and variable data
/tmp	Temporary files



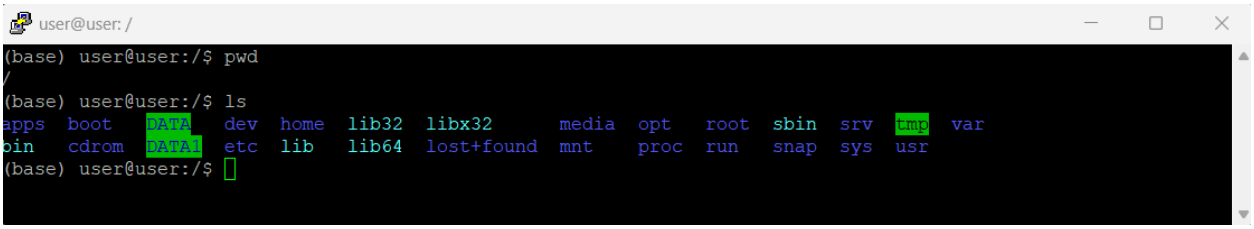
Basic Shell Commands

3.1 Navigating the Filesystem

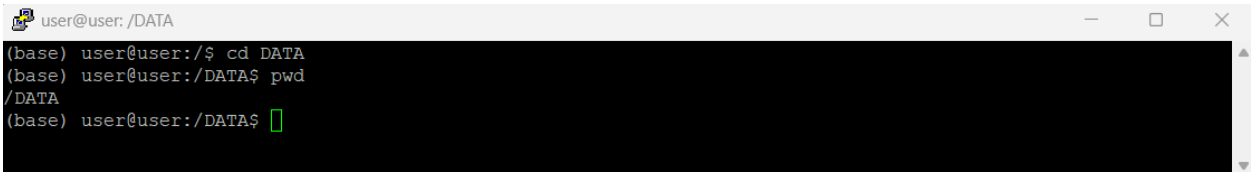
pwd # Print working directory



ls # List files



cd /path # Change directory



Try following: ls, ls-l, ls-al, ls-lrt, ls *.c

3.2 File Operations

touch file.txt # Create empty file

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ pwd
/DATA/clab
(base) user@user:/DATA/clab$ ls
(base) user@user:/DATA/clab$ touch file.txt
(base) user@user:/DATA/clab$ ls
file.txt
(base) user@user:/DATA/clab$
```

mkdir dir # Make directory

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ mkdir dir
(base) user@user:/DATA/clab$ ls
dir  file.txt
(base) user@user:/DATA/clab$
```

cp src dst # Copy file/directory

```
user@user: /DATA/clab/dir
(base) user@user:/DATA/clab$ pwd
/DATA/clab
(base) user@user:/DATA/clab$ ls
dir  file.txt
(base) user@user:/DATA/clab$ cp file.txt dir/
(base) user@user:/DATA/clab$ cd dir
(base) user@user:/DATA/clab/dir$ ls
file.txt
(base) user@user:/DATA/clab/dir$ pwd
/DATA/clab/dir
(base) user@user:/DATA/clab/dir$
```

mv old new # Move/rename

```
user@user: /DATA/clab/dir
(base) user@user:/DATA/clab/dir$ pwd
/DATA/clab/dir
(base) user@user:/DATA/clab/dir$ ls
file.txt
(base) user@user:/DATA/clab/dir$ mv file.txt file2.txt
(base) user@user:/DATA/clab/dir$ ls
file2.txt
(base) user@user:/DATA/clab/dir$
```

rm file.txt # Delete file

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ ls
dir  file.txt
(base) user@user:/DATA/clab$ rm file.txt
(base) user@user:/DATA/clab$ ls
dir
(base) user@user:/DATA/clab$
```

vi hello.c # opens vi editor

vim file.txt

i - insert mode, ESC - command mode, :wq - save and exit

{Note: for working on vi editor, before writing in the editor you have to press i before writing, once writing is done save it by pressing Esc then write :wq}

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ ls
dir
(base) user@user:/DATA/clab$ vi hello.c
(base) user@user:/DATA/clab$ ls
dir  hello.c
(base) user@user:/DATA/clab$
```

3.3 Viewing Files

cat file.txt # Display content

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ cat hello.c
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
(base) user@user:/DATA/clab$
```

less file.txt # Scrollable view

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ less hello.c
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
~
~
```

head -n 10 file.txt # First 10 lines

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ head -n 3 hello.c
#include <stdio.h>

int main() {
(base) user@user:/DATA/clab$
```

tail -n 10 file.txt # Last 10 lines

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ tail -n 3 hello.c
    printf("Hello, World!\n");
    return 0;
}
(base) user@user:/DATA/clab$
```

gcc -o hello hello.c # compiling a c code in gcc compiler

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ ls
dir  hello.c
(base) user@user:/DATA/clab$ gcc -o hello hello.c
(base) user@user:/DATA/clab$ ls
dir  hello  hello.c
(base) user@user:/DATA/clab$ ./hello
Hello, World!
(base) user@user:/DATA/clab$
```

Try following: **cat filename.txt, more filename.txt, less filename.txt, touch file.txt, cp a.txt b.txt, mv a.txt b.txt , rm file.txt ,rm -i file.txt, rm -rf folder/ ,find . -name "*.c" , head -n 5 file.txt, tail -n 5 file.txt , wc -l file.txt, wc file.txt**

4. File Permissions and Ownership

4.1 Understanding Permissions

ls -l

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ ls -l
total 28
drwxrwxr-x 2 user user  4096 Aug  3 10:52 dir
-rwxrwxr-x 1 user user 16696 Aug  3 12:19 hello
-rw-rw-r-- 1 user user    80 Aug  3 12:10 hello.c
(base) user@user:/DATA/clab$
```

 # rwxr-xr-- => User (rwx), Group (r-x), Others (r--)

Modifying Permissions

chmod 755 file.sh # Read-write-execute for user, read-execute for others

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ ls -l
total 28
drwxrwxr-x 2 user user  4096 Aug  3 10:52 dir
-rwxrwxr-x 1 user user 16696 Aug  3 12:19 hello
-rw-rw-r-- 1 user user    80 Aug  3 12:10 hello.c
(base) user@user:/DATA/clab$ chmod 756 hello.c
(base) user@user:/DATA/clab$ ls -l
total 28
drwxrwxr-x 2 user user  4096 Aug  3 10:52 dir
-rwxrwxr-x 1 user user 16696 Aug  3 12:19 hello
-rwxr-xrw- 1 user user    80 Aug  3 12:10 hello.c
(base) user@user:/DATA/clab$
```

chown user: group file # Change ownership

```
user@user: /DATA/clab
(base) user@user:/DATA/clab$ ls -l
total 28
drwxrwxr-x 2 user user  4096 Aug  3 10:52 dir
-rwxrwxr-x 1 user user 16696 Aug  3 12:19 hello
-rwxr-xrw- 1 user user    80 Aug  3 12:10 hello.c
(base) user@user:/DATA/clab$ chown ishwar_2221cs30 hello.c
chown: changing ownership of 'hello.c': Operation not permitted
(base) user@user:/DATA/clab$ sudo chown ishwar_2221cs30 hello.c
[sudo] password for user:
(base) user@user:/DATA/clab$ ls -l
total 28
drwxrwxr-x 2 user            user  4096 Aug  3 10:52 dir
-rwxrwxr-x 1 user            user 16696 Aug  3 12:19 hello
-rwxr-xrw- 1 ishwar_2221cs30 user    80 Aug  3 12:10 hello.c
(base) user@user:/DATA/clab$
```

Task 1: Create, compile, and run a simple C program in Linux, start by creating a file named hello.c using a text editor like nano with the command nano hello.c. Inside the file, write the following code:

```
#include <stdio.h>

int main() {    printf("Hello, World!\n");    return 0;}
```

Save the file by pressing Ctrl + O, then Enter, and exit with Ctrl + X. To compile the program, use the GNU Compiler Collection (GCC) by running the command gcc hello.c. This will produce an executable named a.out by default. You can then run the compiled program by executing ./a.out in the terminal, which will display the output Hello, World!. If you prefer to name the output file something else, you can compile it using gcc hello.c -o hello and then run it with ./hello

Task 2: Record the important Linux commands you have learned in a tabular (column) format, including a brief description of each command.

For Eg.

ls -lrt Lists files in long format, sorted by modification time (oldest first).

ls *.c Lists all files with a .c extension.

chmod -r file.txt Removes read permission

wc -m file.txt Character count

wc -L file.txt Length of the longest line

Task 3:

- Modify and test the 'Hello World' C program by incorporating a system command
- Run basic shell commands like `date`, `whoami`, `pwd` from within a C program.
- Compile another C file using `system("gcc file.c -o file");` from a C program.

```c
int main() {

    printf("Hello, World!\n");

    // Run a system command (e.g., list directory contents)
    printf("Running 'ls -l' command:\n");
    system("ls -l");

    return 0;
}
```

Ta